

JOHNSON GRANT  
IN-61-CR  
312542

(NASA-CR-187404) AdaNET PROTOTYPE LIBRARY  
ADMINISTRATION MANUAL (GHG Corp.) 22 P  
CSCL 098

N91-13101

Unclassified  
63/61 0312542

# **ADANET PROTOTYPE LIBRARY**

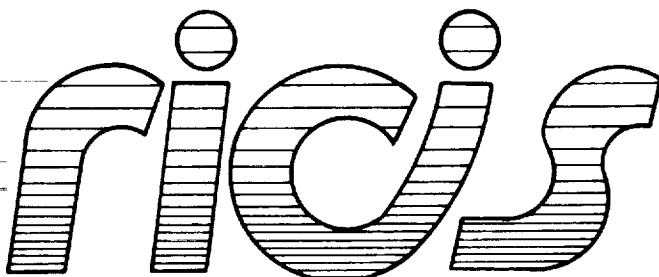
## **Administration Manual**

**Lionel Hanley**  
*GHG Corporation*

**May 1989**

**Cooperative Agreement NCC 9-16**  
**Research Activity No. SE.24**

**NASA Headquarters**  
**Technology Utilization Program**  
**Information & Network Operations**



*Research Institute for Computing and Information Systems*  
*University of Houston - Clear Lake*

**T·E·C·H·N·I·C·A·L      R·E·P·O·R·T**

## *The RICIS Concept*

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

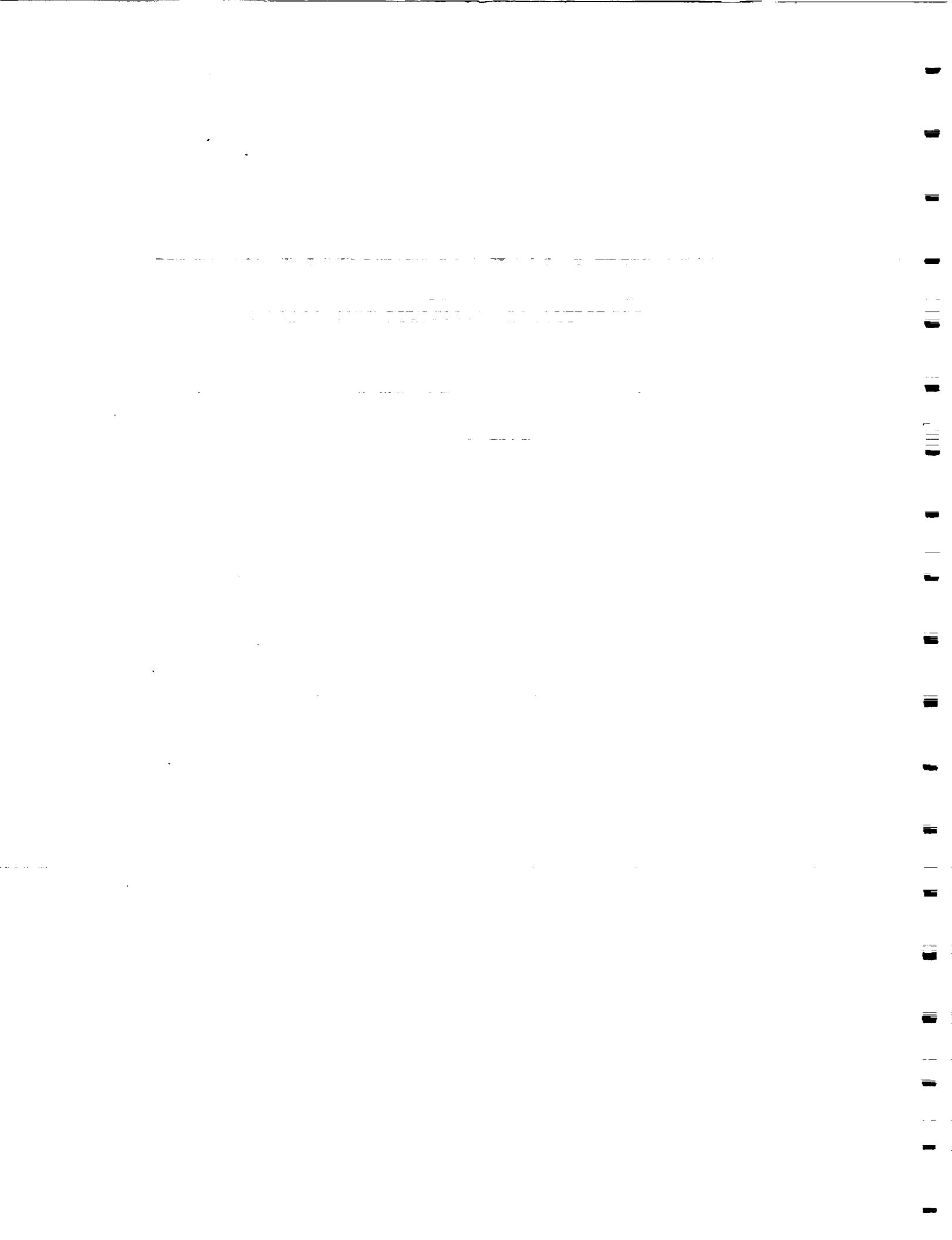
Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

# ***ADANET PROTOTYPE LIBRARY***

## ***Administration Manual***

**Lionel Hanley**  
***GHG Corporation***



## **Preface**

This research was conducted under the auspices of the Research Institute for Computing and Information Systems by Lionel Hanley of GHG Corporation. Dr. Charles McKay, Director of SERC, at the University of Houston-Clear Lake served as RICIS technical representative.

Funding has been provided by NASA Technology Utilization Program, NASA Headquarters through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Roy Bivins, Manager, Information and Network Operations, Technology Utilization Division, NASA Headquarters.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.



# AdaNet

## PHASE 0 SUPPORT FOR THE ADANET DYNAMIC SOFTWARE INVENTORY (DSI) MANAGEMENT SYSTEM PROTOTYPE

Project No. RICIS SE. 24

### AdaNet Prototype Library Administration Manual

---

**Research Institute for Computing  
and Information Systems (RICIS)  
Cooperative Agreement NCC-9-16**

**Revision #01  
May 1989**

**University of Houston Clear Lake  
2700 Bay Area Blvd  
Houston, Texas 77058-1068**

**GHG SE 24-2**

**ADANET**  
**AdaNet Prototype Library Administration Manual**

**Prepared by**  
**GHG Corporation**  
**Houston, Texas**

**Under**  
**Subcontract No. 031**

**for**  
**Research Institute for Computing and**  
**Information Systems (RICIS)**  
**University of Houston Clear Lake**  
**Houston, Texas**

**MAY 1989**

**APPROVED BY GHG:**

---

**G. O'Neal, Software Engineer**  
**GHG Ada Technology Group**

---

**L.G. Hanley, Manager**  
**GHG Ada Technology Group**

## Table of Contents

THE ADANET LIBRARY OF REUSABLE SOFTWARE PARTS .....	3
1.0 INTRODUCTION .....	3
2.0 OVERVIEW AND CONCEPTS .....	3
2.1 The Library Concept .....	3
2.2 Hierarchy .....	4
2.3 Architecture .....	5
2.3.2 Single Project Library View .....	5
2.4 Library Membership .....	5
2.5 Trust Vector .....	5
2.6 Standardization Across Libraries .....	5
3.0 INFORMATION STORED IN THE LIBRARY .....	6
4.0 TRUST VECTOR .....	6
4.1 Library Part Constituent Completeness .....	7
4.2 Testing Completeness Criteria .....	7
4.3 Usage Metrics .....	8
4.4 Static Quality Metrics .....	8
4.5 Reusability Metrics .....	8
5.0 SEARCH AND RETRIEVAL .....	8
5.1 Part Classification .....	9
5.2 Part Cataloging .....	9
5.3 User Interface to the Cataloging/Retrieval Mechanism .....	9
5.4 Additional Automated Support .....	9
5.5 Library Access Responsiveness .....	10
6.0 COMMUNICATION .....	10
6.1 Error Reports .....	10
6.2 Feedback Reports .....	11
6.3 User-to-User Communications .....	11
7.0 CONFIGURATION MANAGEMENT .....	12
7.1 Configuration Control .....	12
7.2 Change Control .....	12
7.3 Error Reporting and Suggested Upgrades .....	12
8.0 LIBRARY USAGE FROM THE RETRIEVER'S PERSPECTIVE .....	12
8.1 Documentation .....	12
8.2 Access .....	12
8.3 Searching .....	12
8.4 Retrieval Procedures .....	12
8.5 Feedback from Retrievers .....	12
8.5.1 Usage Reporting .....	12
8.5.2 Discrepancy Reports .....	12
9.0 LIBRARY USAGE FROM THE SUBMITTER'S PERSPECTIVE .....	12
9.1 Submitter Qualifications .....	12
9.2 Submission Processing .....	12
9.2.1 Submission Procedures .....	12
9.2.2 Submission Information .....	12
9.3 Acceptance Procedures .....	14
10.0 LIBRARY ADMINISTRATION .....	14
10.1 Library Policy .....	14
10.2 Library Operation .....	15
10.2.1 Storage Format .....	15
10.2.2 Physical Access .....	15
10.3 Discrepancy Reports .....	16

## THE ADANET LIBRARY OF REUSABLE SOFTWARE PARTS

This document describes the functions of the AdaNet Prototype Library of Reusable Software Parts. Adopted from the Navy Research Laboratory's *Reusability Guidebook* (V.5.0), this is a working document, customized for use by the AdaNet Project.

Within this document, the term "part" is used to denote the smallest unit controlled by a library and retrievable from it. A part may have several constituents, which may not be individually tracked.

Presented in this document are:

- \* The types of parts which may be stored in the library and the relationships among those parts
- \* A concept of trust indicators which provide measures of confidence that a user of a previously developed part may reasonably apply to a part for a new application.
- \* Search and retrieval, configuration management and communications among those who interact with the AdaNet Prototype Library
- \* The AdaNet Prototype, described from the perspective of its three major users: the part reuser and retriever, the part submitter, and the librarian and/or administrator.

The first draft of this document is intended to give guidance to the developers of the AdaNet prototype. As the AdaNet System progresses, this manual should be modified to reflect the current operation of the AdaNet Library system.

### **1.0 INTRODUCTION**

Many types of libraries exist, informal and formal, small and large, reference and working. Using the Clear Lake Model, the architecture of the AdaNet Prototype Dynamic Software Inventory (DSI) Management System can either be simplified to describe a small, tightly-focused project library or extended to describe a national library like AdaNet.

Because a wide variety of libraries will be useful to various communities, the library mechanism or infrastructure does not constrain the library contents or uses. Thus a single language library or narrow-application domain library can be supported by the same architecture as a national, general purpose library. The policies for the AdaNet Prototype will be occasionally noted when they are significant.

### **2.0 OVERVIEW AND CONCEPTS**

#### **2.1 The Library Concept**

The library serves as a focal point for the collection and distribution of software parts. To provide adequate response to inquiries, it must contain descriptive and evaluative information relevant to the user community. The more focused the user community, the more detailed will be the evaluative information. For example, corporate libraries may require elaborate specification of Quality Assurance procedures used for each part, while national libraries might only cite relevant NASA or DoD standards.

Libraries face practical startup problems in acquiring a useful collection of parts and attracting a user community. Therefore, the library mechanism must support a heterogeneous collection of library parts but evolve toward libraries of parts carefully designed for reuse and thoroughly tested for reliability.

Currently, the RSL contains public domain Ada source code and documentation that may be useful to the AdaNet project. Reusable parts developed for the AdaNet Prototype will be thoroughly tested then added to the RSL. Once the AdaNet Library Prototype is functional, those trusted parts in the RSL will be transferred into the prototype. Thus the capabilities of the AdaNet prototype will be demonstrated with reusable parts developed by the prototyping effort.

The architecture permits manual as well as highly-automated methods.

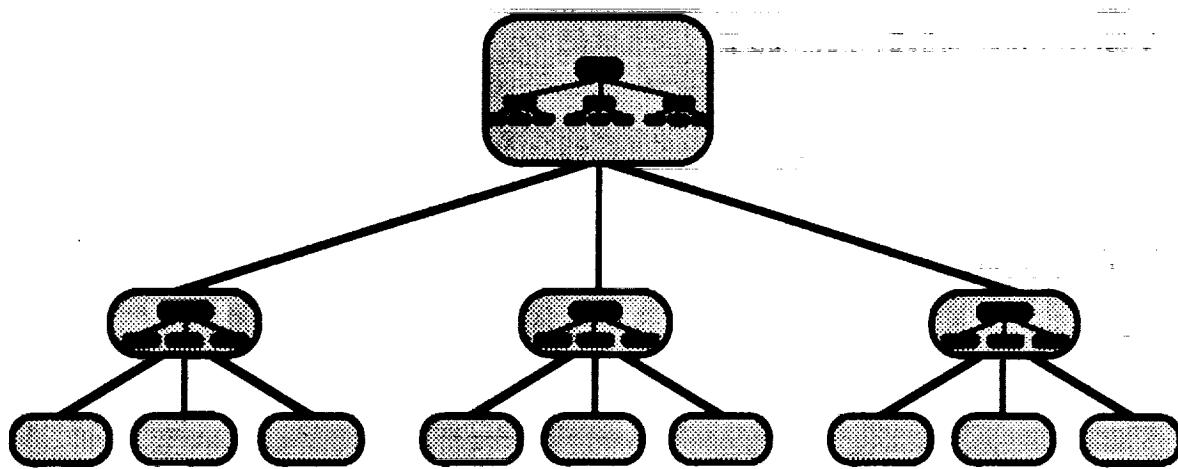
Specialized libraries can impose elaborate constraints on parts and their relationships.

## 2.2 Hierarchy

A hierarchy of libraries can be established nationally, like a book library with a main library and branch libraries. Each local library will specialize in a particular domain of software engineering and will maintain a repository of those components of interest to the local users. Every branch library will have access to the resources of every other branch through the main library or directly with each other.

All libraries will be architecturally similar to allow information to flow easily between them. Each library will contain a repository, a catalog, a bulletin board and software acceptance and distribution processes. Libraries that are high in the hierarchical scheme, like the AdaNet main library, will be used primarily to search or locate parts. They will contain complete indexing and searching capability and sufficient abstract type information to permit refinement of search requests. Bulk information pertaining to a part (such as design, code, etc.) need not be directly available, but the actual location of such information must be known.

Libraries further down the hierarchical scheme may typically contain parts for a limited user community. At the bottom of the hierarchical scheme would be libraries that contain the reusable parts required by the designer. The scheme is depicted in Figure 2.2-1. For example, using a manufacturing industrial model, the Level 1 library could be a large, government agency or corporate field office. The Level 3 library would be the working library used actively by the designers of a specific project.



**Figure 2.2-1.**

If several groups are working on similar projects, they may provide each other with access to their local libraries. Although a single, common library might be preferable, it might not be possible (e.g., if the groups are geographically separated or if the groups work for different companies).

Groups of libraries can be established to serve a variety of needs. These libraries might be available within a single company, among a group of companies, or between companies and the government.

These considerations imply that libraries can easily exchange parts and catalogs. Within the AdaNet Library System, an interchange protocol must be defined. This AdaNet interchange protocol should include:

- \* Media formats
- \* Parts and catalog
- \* Parts usage and quality metrics data
- \* Library configuration and indexing data

If library implementations are not identical, then means for information exchange must be defined. A common import/export format may be identified or loader programs may be written to share parts and catalogs between AdaNet and other library systems and repositories.

### 2.3 Architecture

This section describes an architecture capable of supporting the variety of libraries noted in 2.2. The architecture assumes that an implementation may be centralized or distributed. It may co-locate the catalog/indexing information with the bulk part information, or separate them. It may be a manual or highly-automated process.

In all cases, a library mechanism must include an acceptance process, a storage and cataloging process, and an access/retrieval process.

acceptance process  
storage and cataloging process

access/retrieval process

### **2.3.2 National Library**

TBD

AdaNet,DoD, NASA, COSMIC,...

### **2.3.2 Single Project Library**

During the development process of a project, several reference libraries could provide search and retrieve facilities to designers and developers. Software that could be reused for the project could be retrieved from all of these reference libraries and placed in working libraries that are under project control. Although some library parts could be used without modification, library parts needing modification would undergo the same software development procedures and controls as newly developed software. When the project is finished, modified library parts and new software with a high potential for re-use should be submitted to the appropriate library for inclusion.

## **2.4 Library Membership**

In order for a user to obtain access to a library or to submit information to the library, library membership must be required. Membership criteria among libraries may vary slightly, but must include an unambiguous, reliable identification of the user. Membership in a national library may require the sponsorship of a government agency, while membership in a corporate library may depend on the department or project assignment. Some libraries may have multiple membership classifications or require payment for services. Customized membership services may include automatic notification of revisions or error reports or early notification of new accessions.

AdaNet membership fee schedule....TBD

## **2.5 Trust Vector**

A series of trust indicators should be maintained for each library part. The trust indicator is a vector of metrics which enables the library user to measure confidence in the library part. The trust level vector includes:

- \* Part constituent completeness
- \* Testing completeness
- \* Usage metrics
- \* Static quality metrics
- \* Reusability metrics

## **2.6 Standardization Across Libraries**

Standards for cataloging must be established. The catalog must contain a basic set of part attributes sufficient enough for the user to make an informed choice about a part.

These attributes include a brief description (abstract) of the part, constituents of the part (e.g., design, document, code), domain of application, language used, and some

reusability metrics, if available. Specific attributes should be defined and made uniform within each specific domain.

### **3.0 INFORMATION STORED IN THE LIBRARY**

This section provides an overview of the data that should be contained in a reusable software library. Three types of information are stored for each part:

- \* Part constituents
- \* Part attributes
- \* Part relationships

A part constituent is any of the following:

- \* Abstract
- \* Requirement specification
- \* Functional specification
- \* Design specification
- \* Algorithmic specification
- \* Source code
- \* Object code
- \* Test specification
- \* Test source code (scripts, build & make files etc.)
- \* Test data
- \* Maintenance/Operations/User manuals
- \* Training material

Part attributes are more fully discussed in Section 9.2 but include such items as:

- \* Submitting organization
- \* Languages
- \* Trust indicator
- \* Reusability metrics

Part relationships are maintained by the library but usually supplied by the submitter. Such data include:

- \* Derived from (child of)
- \* Spawned (parent of)
- \* Variant of (name and version of sibling)
- \* Used by
- \* Uses

### **4.0 TRUST VECTOR**

Trust indicators are selected metrics assigned to each library part. The Trust Vector gives the AdaNet user an indication of the quality of a part.

Five different trust areas have been identified:

- \* Library part constituent completeness
- \* Reliability and test history

- \* Reusability metrics
- \* Quality metrics
- \* Usage metrics

#### 4.1 Library Part Constituent Completeness

##### LEVEL OF COMPLETENESS

<u>Constituent</u>	<u>Completeness</u>
Abstract	10
Requirement specification	10
Functional specifications	10
Design	10
Algorithm/function	10
Source code	10
Object code	10
Test specification	10
Test code	10
Test data	10
Maint/operations/user manual	10
Training materials	10
<hr/>	
Total Completeness Value	10 * (total # constituents)

#### 4.2 Testing Completeness Criteria

For each part, there is an associated testing completeness indicator.

- Compiled
- Loaded
- Ran
- Unit-tested
- Tested in a laboratory environment
- Integrated with other parts
- Integration-tested
- Exhaustive test (most of paths tested)
- Released
- Site-tested
- In an operational environment
- User reported bugs have been corrected
- List of bugs exists
- Detailed test history exists
- Test plan exists
- Detailed test history exists
- Test report exists
- Used by a person/organization other than the author
- Used in a production or end-user environment

#### 4.3 Usage Metrics

For each non-commercial library part, a set of usage metrics is maintained. These metrics contain the following components:

- \* Number of times retrieved
- \* Number of times used with modification
- \* Number of times used without modification
- \* Number of times retrieved but not used

#### 4.4 Static Quality Metrics

Static Quality Metrics are metrics, applied to source code by manual or automated means, used to determine how well software has been designed and whether it is usable and maintainable. Integration of COTS automated test tools like AdaMAT, ATVS should be considered.

#### 4.5 Reusability Metrics

The reusability metric is a measure of the degree that reusable software adheres to engineering principles. These principles form the basis for the following reusability metric:

Part usable within an application area	10
Part usable across application areas	10
Selected paradigm supports reusability	10
Designed with protection and minimization of interfaces	10
Designed with information hiding	10
Designed independent of the environment	10
Designed for portability	10
Designed for maximum cohesion	10
Designed for minimum coupling	10
	---
	90

### 5.0 SEARCH AND RETRIEVAL

Search and retrieval support will be engineered to meet the following objectives:

- \* Rapid identification of those catalogued parts that most closely match the retriever's specification (see 6.2.6 Standardization Across Libraries).
- \* Friendly computer-based assistance to the reuser in the search process (consider background daemons matching components to requirements.)
- \* Flexible search processes that allows the retriever to prioritize search goals (e.g., trust level)
- \* A wide-spectrum extendible sets of classification attributes for precise part characterization and identification
- \* Retrieval in a variety of alternative formats (e.g., abstracts, full documentations, source code, etc.)

#### 5.1 Part Classification

Classification in general is the act of grouping similar objects such that all members of a group or class share at least one characteristic that members of other classes do not possess. The result of classification is a network or structure of relationships. A classification

scheme is a tool for the production of systematic order based on a controlled and structured index vocabulary. This index vocabulary is called the classification schedule. It consists of the sets of names or symbols representing concepts or classes, listed in a systematic order to display the relationship between classes. These names or symbols are called terms.

The classification scheme should be:

- \* Flexible--independent of any one particular hierarchical view
- \* Extensible--able to accept new terms without creating problems in re-classification
- \* Adaptable--amenable to different database organizations

Because synonyms can produce different characterizations for the same part, a thesaurus is needed for vocabulary control. Terms from existing classification schedules include: Booch, EVB GRACE, Pat's thesis, ACM Computing Reviews, AFIPS taxonomy, AIAA, IMSL, RAPID, Reuben Prieto-Diez.

## **5.2 Part Cataloging**

Cataloging consists of selecting a set of attributes that characterize a particular object for a specific group of users. A set of standard cataloging guidelines

## **5.3 User Interface to the Cataloging/Retrieval Mechanism**

Because the library will be utilized by librarians and retrievers with differing levels of experience and skill, the user interface should present the information contained in the library in a variety of user-selectable formats (e.g., menus, graphics, etc.) Users should be able to request on-line assistance in retrieving or specifying library information. A complete hard copy of the library catalog and usage instructions should also be available.

Because some library parts, documentation or associated information cannot be stored or distributed electronically (e.g., graphs, pictures, charts, etc.), the library interface should announce that non-electronic documents exist and can be requested through other means.

## **5.4 Additional Automated Support**

The use of additional automation can aid in the management of the parts library and its classification scheme. Automated support may include:

- \* Software analysis tools for the uniform acquisition of metrics used in assessing ease of part modification
- \* Software analysis tools for automatic application of discriminator values (index terms)
- \* Expert assistance tools for browsing, reducing the search effort and increasing the precision of matches
- \* Measurement/Evaluation tools to collect information from successful and unsuccessful queries to detect needed refinements in the index set, identify new parts to be developed and improve the user interface to the library

## **5.5 Library Access Responsiveness**

Catalog search time and actual part retrieval time are important to the user. Both must be considered when designing the library.

Interactive access should be provided for searching the parts catalog. This will enable the user to quickly search a particular library and (ideally) be able to find the needed part in one session.

Less critical than rapid search of the catalog is the minimal time for part retrieval. Each library should be able to indicate the typical time needed to retrieve a part associated with the retrieval process.

Users can become discouraged from using the library by lengthy delays experienced in search and retrieval. On the other hand, high-speed search and retrieval for larger libraries would be impractical. Bearing in mind that each library user base has different needs, the library administrator must determine what the time restraints should be for each access method.

## **6.0 COMMUNICATION**

In order to promote the exchange of information about the parts within the library, it will be necessary to administer mechanisms to:

- \* Report errors
- \* To report experience with a part and requests for enhancements and modifications
- \* To permit user-to-user communication

### **6.1 Error Reports**

The library should provide a way to receive reports concerning errors or problems and to communicate those reports to the supporting organization. Although the library should not be responsible for verifying the accuracy of these reports, it should make a "good faith" effort to communicate reports of calamitous errors to other part retrievers. Specifically, the library should:

- \* Allow retrievers to specify whether they want to be notified about subsequent changes or errors in the part
- \* Receive and log problem reports from users of the library parts
- \* Forward problem reports to the support organization (if one has been identified), with attention given by library personnel to eliminate duplicate reports
- \* Notify past retrievers about problems reported with the parts they have received
- \* Notify potential users of problem reports associated with parts they are considering

The library should be able to remove problem reports against parts when:

- \* The support organization or an individual submits a correction
- \* The problem report is identified as an error made by the user
- \* The problem report does not identify legitimate issues (e.g., it is designed to influence the incentives for reuse)

## **6.2 Feedback Reports**

The library should collect information to measure the effectiveness of the library and to communicate information about parts from the users to the maintenance group(s). The effectiveness of the library can be measured using:

- \* Part usage metrics
- \* The number of error reports
- \* Other statistics the librarian deems necessary

The library will also receive reports from users that indicate:

- \* Whether the user actually used the retrieve part
- \* Whether the part was used "as-is" or as a basis for departure
- \* Whether the retrieved part was effective or useful
- \* Whether enhancements or modifications are requested

This information should be periodically reported to the library part's maintaining organization (if any).

## **6.3 User-to-User Communications**

The library should maintain a means, such as bulletin boards, for user-to-user and user-to-group communications. The system should also include a mechanism to maintain the anonymity of senders and receivers, since some users may be discouraged from contributing unless they can do so anonymously.

User-to-user and user-to-group communications should be available in packet communications form, such as electronic mail or via the postal service. The library should also support a means, such as a bulletin board, to facilitate communication among users. This support should:

- \* Be accessible to all members of the library
- \* Be topic oriented (in effect, serving as a user's group independent of anyone else)
- \* Cover the library operation itself
- \* Poll users sharing a common interest
- \* Provide a thesaurus of topics
- \* Facilitate entry of library items
- \* Tag entries with an alias so library staff can trace authors of entries
- \* Contain expiration dates so entries can be kept current

In each case, there must be a mechanism to maintain the user's anonymity, as desired by the receiver of sender of a message.

## **7.0 CONFIGURATION MANAGEMENT**

**TBD**

configuration control and change control

### **7.1 Configuration Control**

**TBD**

## **7.2 Change Control**

**TBD**

## **7.3 Error Reporting and Suggested Upgrades**

**TBD**

# **8.0 AdaNet USAGE FROM THE RETRIEVER'S PERSPECTIVE**

## **8.1 Documentation**

**User's Guide**

**On-line help**

**TBD**

## **8.2 Access**

**Network access**

**dial-in**

**TBD**

## **8.3 Searching**

**TBD**

Pattern matching daemons (McKay) could be employed as background processes within the AdaNet Library System. For example, these daemons could automatically match requirements to part specifications as the requirements are being entered into a project library. A list of parts that may be reusable would then be available during requirements analysis. Thus requirements could be modified to maximize reuse early in development.

## **8.4 Retrieval Procedures**

**TBD**

## **8.5 Feedback from Retrievers**

**TBD**

### **8.5.1 Usage Reporting**

**TBD**

- \* make support organizations aware of possible problems or generalizations
- \* update the part trust level
- \* identify users that might need error alerts and fixes
- \* help identify new parts that are needed

### **8.5.2 Discrepancy Reports**

Retrievers report errors encountered in the use of retrieved library parts.

## **9.0 AdaNet USAGE FROM THE SUBMITTER'S PERSPECTIVE**

TBD

### **9.1 Submitter Qualifications**

TBD

### **9.2 Submission Processing**

TBD

#### **9.2.1 Submission Procedures**

TBD

#### **9.2.2 Submission Information**

TBD

#### **PART DESCRIPTION**

Title

Type of Part (code, design, etc.)

Type of function

Purpose of Function

Interface requirements-required information about the  
software and hardware not included with the part

#### **SUBMITTER DATA**

Name

Address/Network Address

Phone

Contact

#### **PART CONSTITUENTS**

Abstract

Requirement Specification

Functional Specification

Design

Algorithm or Function

Source Code

Object Code

Test Specification

Test Code

Test Data/Results

Maintenance/Operations/User's Manual(s)

#### **PART HISTORY**

Reason for Part Development

Date of Completion

Description of Applications Used

Frequency of Use

Description of Development Standards

Version Number

**PART RELATIONSHIPS**

Name of Parent [Originally-derived from (child)]

Name of Children [Originally-spawned from (parent)]

Name and version of Siblings

Used by [Originally-derived from same part]

Uses

**PART ATTRIBUTES**

Keywords (to search/retrieve on)

Development Language

Host Environment (Computer and Operating System)

Target Environment (Computer and Operating System)

**RESTRICTIONS**

Government

Developer

Environment Imposed (compiler, tools, peripherals,...)

Reusability Metrics (see Section 4.5)

**DISCLAIMERS**

Warnings

Problems

Limitations

Lack of Tests

**SOFTWARE SUPPORT**

Support Organization or Person

Qualification

Frequency of Update

**MISCELLANEOUS INSTRUCTIONS**

How to Get Part

Fees

Warranties

**RELEASES**

Transfer and/or Assignment of Copyright

Transfer of Ownership of Hard Goods

**DELIVERABLE MEDIA DESCRIPTION**

Media

Hard Goods in Delivery

**MEDIA**

Electronic

Magnetic

Optical

Paper

Type of Format (e.g. ASCII record, etc.)

### **9.3 Acceptance Procedures**

TBD

## **10.0 LIBRARY ADMINISTRATION**

### **10.1 Library Policy**

Commercial software shall not reside in the library. Information describing commercial software and contact data for vendors/agencies holding it are appropriate. The library should not be responsible for maintaining the privacy/security of commercial/proprietary information nor for distribution thereof. However, in the interest of facilitating software reuse, information about reusable commercial software may be included.

Most libraries will not contain classified information, but may contain pointers and instructions for acquisition of classified information. Use of an unclassified library should not preclude assisting in the reuse of classified software products when feasible. A classified library could be established using the same framework as the unclassified one (e.g., run it at a "system-high" level).

The library will attempt to report problems to all users of a part in a timely manner. The problems may be errors or other areas of concern. Care must be taken to assure a real problem has occurred, not operator error or misunderstandings.

The library shall promote its use to an appropriate user base.

Charges and fees associated with the library should be consistent with the long range objectives for better quality and low cost software. The library should be free or at low cost (perhaps funded in prototype by the government or corporate body). Once it is established, it can be made self-sustaining.

Most libraries will not be responsible for maintenance of stored parts. Maintenance will be the responsibility of the submitter, the support organization or other designee.

A trust vector shall be assigned to each part in the library. Typically, the best vector shall represent a part that is fully tested against approved quality metrics and has proven usage. The lowest vector may represent "buyer beware," but have proven potential for applicability.

The author's name typically will not be described to users nor will users be identified to the author for routine matters. Generally, the librarian will serve as the intermediary.

Ada shall be the predominate language supported by the library, although other documents including requirements and designs description will be supported as well.

## **10.2 Library Operation**

### **10.2.1 Storage Format**

All participating AdaNet libraries must be capable of producing media in the AdaNet interoperability format.

### **10.2.2 Physical Access**

The AdaNet network configuration is presented in Figure 10.2.2.1. Gateways to AdaNet from the Defense Data Network (DDN) and NASA PSCN are listed in Figure 10.2.2.2.

**TBD**

**Figure 10.2.2.1.**

**TBD**

**Figure 10.2.2.2.**

### **10.3 Discrepancy Reports**

AdaNet users may report errors encountered in the use of library system via an online discrepancy report form available on the bulletin board system or by mail to AdaNet.

**# # #**